Weighted Automata, Constraint Programming, and Large Neighborhood Search for the PATAT 2010 – NRP Competition

Julien Menana[1] and Sophie Demassey[1⋆]

École des Mines de Nantes, LINA CNRS UMR 6241, F-44307 Nantes, France.
{julien.menana,sophie.demassey}@mines-nantes.fr

A solution approach for the Nurse Rostering Problem NRP10 proposed at the PATAT 2010 competition is presented. The approach makes cooperate three core elements: (1) weighted automata to easily model and combine the scheduling rules, (2) constraint propagation to process the automata together with the transversal cover requirements, (3) an incomplete large-neighborhood search.

1 Modelling sequencing rules as automaton-constraints.

The `regular` global constraint [2] has brought new perspectives for modelling and solving personnel scheduling problems with Constraint Programming (CP). Given a deterministic finite automaton Π and a finite sequence of discrete variables $S = (S_1, \ldots, S_T)$, constraint $\texttt{regular}(S, \Pi)$ ensures that the sequence of values taken by S is accepted by Π. In personnel scheduling, the schedule of an employee e can be represented as a sequence S^e of activities, where one activity (shift or rest) S^e_t is assigned to each time period (day) $t = 1, \ldots, T$. The schedule is subject to a set of rules, many of them being valid patterns of activities the sequence S^e must comply with. In turn, these patterns can be converted as regular expressions, then bound together as one automaton Π^e. In [1], we proposed a systematic construction of this automaton from a large set of rule types, including the ones considered in NRP10. By extending `regular` to the `multicost-regular` constraint, which handles multi-weighted automata and counters on each weight dimension, we also merged non-regular pattern rules such as the cardinality rules constraining the number of activity occurrences. Experiments showed the benefit to handle simultaneously all the rules into one `multicost-regular` by employee, compared to a separate model, even if arc-consistency cannot be enforced on such a NP-hard optimization constraint.

2 Counting violations using weighted automata.

The construction of the automaton has been adapted for NRP10 to permit rule violations and deal with violation costs. For each rule, a regular pattern and a counter of the occurrence number of the pattern are derived. The violation cost

⋆ with the fruitful collaboration of Benoît Rottembourg, who pointed us out the challenge and inspired many of the present ideas.

depends on this counter and on the rule weight. When this relation is linear, the weight is included into the automaton to directly count the occurrence weight rather than the occurrence number. A `multicost-regular` constraint is then posted for each employee on the intersection of the rule weighted automata. In fine, the constraint accepts any sequences of T activities – since all the rules are soft – and maintains a violation counter or cost variable for each rule. As all assignments are feasible, filtering here may only happen when back-propagating from the cost to the assignment variables. Hence to improve filtering, all linear violation costs are aggregated as one, by simply summing up the automaton transition weights. All rules of NRP10 have linear costs except the number-of-assignment rule. In our experiments, we also set apart the number-of-consecutive-week-ends rule into its own `cost-regular` constraint, in order to keep reasonable the conjugate automaton size (about 200 states, 1000 transitions).

3 Handling the transversal cover requirements.

Orthogonally, a `global-cardinality` constraint [3] is set on each period to model the hard cover requirements. Hence, the assignment variables (S_t^e) are subject to a matrix of constraints with `multicost-regular` on rows/employees and `global-cardinality` on columns/periods. Such decomposition often leads to poor propagation as it does not exploit the inter-relations between rows and columns. A redundant sum constraint coupling the assignment counters of all employees and the cover requests on all periods is helpful there but not enough. To help cover satisfaction, our branching strategy progressively selects the assignment variables in columns. Tree search is also regularly restarted, from a different initial column each time, to add diversity.

4 Minimizing the sum of the violation costs.

The NRP10 model strongly lies on the objective function, given as the sum of the individual violation costs. Again, such a relation leads to poor propagation between the individual cost variables (unless the global sum is close to 0 which seems not to be the case in most of the NRP10 instances). Consequently, the `multicost-regular` constraints may perform few back-propagation. Constraint algorithm computations can be exploited though, at least if not for filtering, externally in the search strategy. In particular, the `multicost-regular`'s compute good estimates of the minimum cost due for each employee-period-activity assignment. These estimates allow to derive for almost free, an efficient regret-based selection heuristic. Given a period t, the regret for an employee e is the difference between the two best estimates among all possible assignments of S_t^e. An employee e with maximum regret is selected and S_t^e is then chosen for the next decision variable to instantiate to its best value.

5 CP-based Large Neighborhood Search.

This branching strategy counter-balances well the model weaknesses as it immediatly finds first solutions of quite good quality. Nevertheless, proving optimality still requires a complete inspection of the search tree. Hence, using a Large Neighborhood Search [4] approach is a good alternative. It is known to be efficient and straightforward to implement on top of an existing CP model and search strategy, as the remaining issue is: given a feasible solution, choose a set of variables to re-assign. Consistent to our model and heuristic, all assignment decisions in a variable number of consecutive columns are dropped, then the same branching strategy is (partially) run to reoptimize.

6 Benefits and drawbacks of the approach.

Our original goal was to develop a generic multi-purpose approach for personnel scheduling, handling many kinds of scheduling rules. This motivated the development of the `multicost-regular` constraint which joins the expressiveness of weighted automata and flexibility of CP. Few adaptations were actually required to our existing approach to handle the rules of the NRP10 challenge. On top of that, this constraint proved to be directly applicable to model any kinds of violation costs. Nevertheless, since all rules may be violated, optimization is preponderant to feasiblity in NRP10, which is a detriment to a pure CP approach. This and the compound objective hinder the propagation abilities of the `multicost-regular` constraints. We developped then an efficient and generic local search strategy made of regret-based heuristic, restarts and repairs. We also focused on more specific techniques (employee symmetric breaking constraints, cost variable branching, etc.) to take into account critical aspects of some instances. Unfortunately, they often interfere with the behavior of the approach in general. In the future, we aim at studying the compatibilities between these techniques, then at integrating them into a self-adaptative approach.

References

1. Julien Menana and Sophie Demassey. Sequencing and counting with the `multicost-regular` constraint. In *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems (CPAIOR'09)*, volume 5547 of *LNCS*, pages 178–192, 2009.
2. Gilles Pesant. A regular language membership constraint for finite sequences of variables. In *Principles and Practice of Constraint Programming (CP'04)*, pages 482–495, 2004.
3. Jean-Charles Régin. Generalized arc consistency for global cardinality constraint. In *Artifical Intelligence (AAAI/IAAI'96)*, pages 209–215, 1996.
4. Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming (CP'98)*, volume 1520 of *LNCS*, pages 417–431, 1998.